

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-022808

(43)Date of publication of application : 26.01.2001

(51)Int.Cl.

G06F 17/50
G01R 31/28

(21)Application number : 11-193869

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 08.07.1999

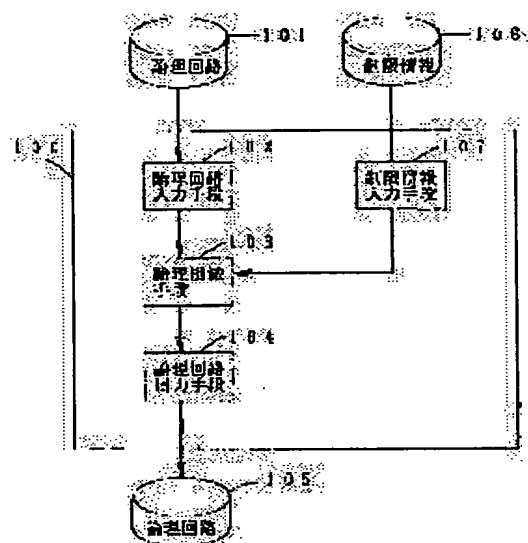
(72)Inventor : MAE YOICHIRO
KAWAMOTO ISAO

(54) LOGIC CIRCUIT REDUCING DEVICE, METHOD AND DEVICE FOR LOGIC SIMULATION

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce logic corresponding to the purpose of verification at a high speed in the logic verification of a large scale circuit.

SOLUTION: Concerning this logic circuit reducing device, a device 100 for converting logic circuits is provided with a means 102 for inputting an object circuit to be verified composed of a logic circuit 101, a limit information input means 107 for supplying limit information 106 corresponding to the verification purpose and a logic reducing means 103 for reducing the inputting object circuit to be verified on the basis of the limit information. The logic simulation method simulates the operation of the logic circuit 101 in which the logic is reduced using this logic circuit reducing device.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

THIS PAGE BLANK (USPTO)

[0042]さらに、第3の制限情報305に注目する。

[0043]制限情報305は変数"sigovf"を識別しないという指定であるため、"sigovf"の値をシミュレーション中に求める必要はない。

[0044]このため、図2の41行の"sigovf"を代入項に持つ代入文を削除する。

[0045]以上の処理により、論理回路図2の記述は図4に変わる。

[0046]なお、ここでは入力とする論理回路として、Verilog-HDLで記述されたRTLレベルの記述を用いた例を用いたが、VHDLやその他の記述言語、また動作レベルやゲートレベルの記述に対しても同様に不要部分の削除を行なえることは言うまでもない。

[0047]以上のように、本発明の第1の実施の形態によれば、請求項1に係る発明である論理回路削減装置によって、シミュレーションの対象となる論理回路から、目的とする検証に不必要な部分を削除することにより、高度なシミュレーションが可能で、回路規模の小さいシミュレーションモデルの記述を得ることができ、図48(第2の実施の形態)第2の実施の形態では、請求項2、3に係る発明に関し図を用いて説明する。図5は請求項2、3に係るシミュレータの構成を示す図である。

[0049]図5において論理回路のシミュレータ500は、被検証対象回路501を入力する論理回路入力手段502と、論理回路に対するシミュレーション実行時の制限情報507を入力する制限情報入力手段508と、検証に不必要な回路をシミュレーション開始前に削除する第1の論理回路削減手段503と、動的な制限情報に基づいてシミュレーション時刻を分割するシミュレーション時刻分割手段509と、分割されたシミュレーション時刻毎に被検証対象回路(論理回路削減回路504)から不要部分を削除してシミュレーションモデルを作成する第2の論理回路削減手段505と、作成されたシミュレーションモデルを用いてシミュレーションを行なうシミュレーション手段508を備えている。

[0050]次に、例を用いて第2の実施の形態に係るシミュレータを具体的に説明する。

[0051]図6はシミュレーション対象となる論理回路により構成されるシステムモデルの最上位階層を表す記述の一部である。

[0052]ただし、図8において、モジュール"filter"は、図2の記述で表された論理回路である。

[0053]図7は図8のシステムモデルに対してシミュレーション時の使用方法について与える制限情報の一例である。

[0054]図7において700は制限を加える信号またはモジュール名である。

[0055]701は加える制限の種類であり、"init"は信号値を固定して用いることを示し、"ignore"はシミュレーション時に値を参照する必要のない信号であることを示し、"remove"はモジュールインスタンスを取り除くことを示している。

[0056]702は信号を固定して用いる場合の信号値であり、"high"は高レベルを固定する時刻に指定されている値に固定することを示している。

[0057]703は制限を加えるシミュレーション時刻を設定する条件、704は制限を解除するシミュレーション時刻を設定する条件であり、"initial"はシミュレーション開始前に制限を加えることを示し、"4000"はシミュレーション時刻"1000"に制限を加えることを示し、"joseph system.mode"、"necode system.mode"はそれぞれ信号"mode"の値が1または0に変化した時点で制限の設定、解除を行なうことを示している。

[0058]図25は、図5の505で実行する論理回路削減方法を示す図である。

[0059]図25において、2501は時刻分割手段508により分割された時刻に与えられた制限の未処理のものがあるかを判定する処理、2502は制限情報507から処理を行なう制限を選択する処理、2503は選択した制限を用いてインスタンス内の論理回路を圧縮する処理、2504は選択した制限に従いインスタンスを削除する処理、2505は削除したインスタンスの出力ポートに接続していた信号線をインスタンスの終了時点で値に設定する処理、2506は時刻分割手段509により分割された時刻に解除する制限の中に未処理のものがあるかを判定する処理、2507は解除する制限を選択する処理、2508は選択した制限を用いて加えたい変更元の状態に直す処理である。

[0080]ただし、2503の論理回路圧縮する処理では第1の実施の形態で示した図24に代い、論理の圧縮を行なう。

[0081]図23は第2の実施の形態に係るハードウェアの構成図の一例を示しており、図23において、2301は処理されたあらゆる情報をみるためのディスプレイ装置、2302は設計者があらゆる情報や処理命令を入力するためのキーボード、2303はあらゆる処理を行う中央演算処理装置、2304は各種情報を格納する記憶装置である。

[0082]第2の実施の形態に係るシミュレータでは、被検証対象回路(図8)と制限情報(図7)が与えられると、制限情報(図7)の中から、シミュレーション開始前に処理可能な静的な制限情報を選び出し、第1の論理回路削減手段503により、第1の実施の形態と同様に回路を削減する。

[0083]図7の例では、703に"initial"の設定がなされ、704には何も指定されていない項目が並び出され、インスタンス"system.filter"に対応するモジュール

によれば、請求項2、3に係る発明である論理回路のシミュレーション方法により、シミュレーションの対象となる論理回路から、目的とする検証に一定の期間不必要な部分を、特定期間削除することにより、目的とする検証に必要なシミュレーションを高速に行なうことが可能となる。

[0075](第3の実施の形態)第3の実施の形態では、請求項4にかかると発明について図を用いて説明する。

[0076]図10は請求項4に係る論理シミュレーション手段の論理回路削減情報を用いる手段の概略的構成図である。

[0077]図10において1001は解析対象のソフトウェア命令系列であるマシン語である。1011は論理回路に与えることができる制限情報である論理回路削減情報であり、1012は可変制限情報を入力する可変制限情報入力手段である。1002は可変制限情報入力手段1012からの情報より、論理回路削減情報108を抽出する制限情報抽出手段である。

[0078]図23は第3の実施の形態に係るハードウェアの構成図の一例を示しており、図23において、2301は処理されたあらゆる情報をみるためのディスプレイ装置、2302は設計者があらゆる情報や処理命令を入力するためのキーボード、2303はあらゆる処理を行う中央演算処理装置、2304は各種情報を格納する記憶装置である。

[0079]次に、例を用いて第3の実施の形態の論理シミュレーション手段を具体的に説明する。図11はシミュレートされるプロセスの命令セットの一例である。命令L Dはアドレスにより指定されるメモリ領域のデータからレジスタAまたはレジスタBにデータを転送するデータ転送命令である。命令S TはレジスタAまたはレジスタBからアドレスにより指定されるメモリ領域にデータを転送するデータ転送命令である。命令M O Vはソースで指定されるレジスタAもしくはレジスタBまたは定数値をデスティネーションで指定されるレジスタAもしくはレジスタBまたはレジスタP S R(プロセッサステータスレジスタ)に代入する算術演算命令である。レジスタP S Rの1ビット目は割込みに対するマスクビットになっており、このビットが1の場合は割込みを受け付けず、0の場合は割込みを受け付ける。スタート時はP S Rの1ビット目は1に設定されている。また、割込みを受け付けた時は自動的にP S Rの1ビット目が1に設定される。命令A D Dはソースで指定されるレジスタAもしくはレジスタBまたは定数値とデスティネーションで指定されるレジスタAまたはレジスタBとを加算し、結果をデスティネーションで指定されているレジスタに代入する算術演算命令である。命令S U Bはソースで指定されるレジスタAもしくはレジスタBまたは定数値をデスティネーションで指定されるレジスタA

ールは図4のように変換され、インスタンス"system.filter"に対応するモジュールは図8のように変換される。

[0084]次に、第1の論理回路削減手段503により作成されたシミュレーションモデルを用いて、シミュレーション手段508に与えられたシミュレーションを行なう。

[0085]この時、第2の実施の形態に係るシミュレータでは、シミュレーション時刻分割手段505により、シミュレーションの実行時に制限情報に指定された条件703または704が成立していないかを監視し、条件が成立している場合、第2の論理回路削減手段505により、被シミュレーション対象のモデルを動的な制限情報に基づいて図25のように変換する。

[0086]図7の例では、4行と7行が動的な制限情報に相当する。

[0087]まず、シミュレーションが図7の第4行の制限を加えるシミュレーション時刻を設定する条件である時刻1000に達し、この時の変数"system.filter2.mode"の値が"high"であるとすると、第2の論理回路削減手段505はインスタンス"system.filter2"に対応するモジュールの記述(図8)の変数"mode"を"high"に固定して第1の実施の形態と同様に制限を削減し、図4のようになっていく。

[0088]そして、シミュレーション手段508は、インスタンス"system.filter2"に対応するモデルとして図8の代わりに図4を用いてシミュレーションを行なう。

[0089]次に、信号sys.modeの値が0から1に変化した場合、図7の第7行の制限に対し、制限を加えるシミュレーション時刻を設定する条件が成り立つため、制限情報に基づきシミュレーション対象(図6)からインスタンス"system.filter2"を取り除いたモデル(図9)を作成し、図8の代わりに図9を用いてシミュレーションを行なう。

[0090]ただし、この時刻に達したインスタンスの出力面については、削除時の値を保持するように設定している。

[0091]さらに、信号sys.modeの値が1から0に変化した場合、図7の第7行の制限に対し、制限を解除するシミュレーション時刻を設定する条件が成り立ち、実行中のシミュレーション対象を図9から図8に戻してシミュレーションを継続する。

[0092]ただし、この時戻されるインスタンス"system.filter2"の内部状態は、削除時の値を用いる。

[0093]なお、第2の実施の形態では制限を加えるシミュレーション時刻の設定を制限情報としてシミュレーション開始前に与えているが、論理回路のシミュレーションに必要となる制限情報を付加して、シミュレーション中に与えることも可能である。

[0094]以上のように、本発明の第2の実施の形態

と論理を削減することにより、シミュレーションを高速化することが可能となる。

【0100】以上説明した様に、請求項1から請求項5に対応する第1から第4の実施の形態の発明によれば、論理回路に検証目的に応じた制限情報を与えることにより、論理回路を削減することが可能である。また、削減された論理回路に対してシミュレーションを行なうことにより、高速に論理検証を行なうことが可能である。また、プロセッサ部とソフトウェアにより制御される論理回路部を持ち、ソフトウェアプログラムコードと論理回路部が相互に作用するシステム全体のシミュレーションにおいて、論理回路部の情報と、ソフトウェアの解析により、検証を行なうソフトウェアから論理回路部の制限情報を作成、論理を削減することにより高速なシミュレーションが可能である。

【0101】

【発明の効果】請求項1記載の論理削減装置によれば、検証目的に応じた制限情報を与え、その制限情報に基づき論理回路から構成される検証対象回路を削減する方法により、検証対象回路を入力し、検証目的に応じた制限情報を入力し、入力した制限情報に基づき、入力された検証対象回路を削減することができる。このため、高速に論理シミュレーションを行なうことができる。

【0102】請求項2記載の論理シミュレーション方法によれば、検証目的に応じた制限情報の各々にシミュレーション時に依存した制限情報を与え、シミュレーション時に依存した制限情報に基づき、シミュレーション時に依存した制限情報を削減し、論理シミュレーション方法により、全体の論理シミュレーションを高速化することができる。

【0103】請求項3記載の論理シミュレーション装置によれば、検証目的に応じた制限情報の各々にシミュレーション時に依存した制限情報を与え、シミュレーション時に依存した制限情報に基づき、シミュレーション時に依存した制限情報を削減し、論理シミュレーション方法により、全体の論理シミュレーションを高速化することができる。

【0104】請求項4記載の論理シミュレーション装置によれば、プロセッサ部を制御するソフトウェアの命令動作を解析し、論理回路部に与えることができる制限情報を入力し、この制限情報とソフトウェアの命令動作の解析結果から、論理を削減する制限情報を抽出し、抽出された制限情報に基づき論理を削減する方法により、論理シミュレーションを高速化することができる。

【0105】請求項5記載の論理シミュレーション装置によれば、検証目的に応じて、プロセッサ部を制御するソフトウェアを指定し、論理回路部に与えることができ、ソフトウェアを指定し、プロセッサ部を制御するソフトウェア命令系列を生成するとともに論理回路部の制限情報を生成し、生成された制限情報に基づき論理を削減する方法により、シミュレーションを高速化することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態における構成を示すブロック図である。

【図2】本発明の第1、第2、第3および第4の実施の形態において例として用いる論理回路をハードウェア記述言語を用いて記述した図である。

【図3】本発明の第1の実施の形態において例として用いる制限情報を示す図である。

【図4】本発明の第1および第3の実施の形態において論理の削減結果として得られる論理回路を示す図である。

【図5】本発明の第2の実施の形態における構成を示すブロック図である。

【図6】本発明の第2の実施の形態において例として用いる論理回路の一部をハードウェア記述言語を用いて記述した図である。

【図7】本発明の第2の実施の形態において例として用いる制限情報を示す図である。

【図8】本発明の第2、第4の実施の形態において動的論理の削減結果として得られる論理回路を示す図である。

【図9】本発明の第2の実施の形態において動的論理の削減結果として得られるシミュレーション対象を示す図である。

【図10】本発明の第3の実施の形態における制限情報抽出の概略概念のプロフロー図である。

【図11】本発明の第3の実施の形態におけるプロセスの命令セットの例を示す図である。

【図12】本発明の第3の実施の形態におけるプロセスサ部と論理回路部とから成る検証対象回路の例を示すブロック図である。

【図13】本発明の第3の実施の形態において実行するプログラムを示す図である。

【図14】本発明の第3の実施の形態における論理回路の可制限情報を示す図である。

【図15】本発明の第3の実施の形態において抽出された論理回路の制限情報を示す図である。

【図16】本発明の第4の実施の形態における論理回路の制限情報の生成および実行プログラム生成の概略概念のブロック図である。

【図17】本発明の第4の実施の形態におけるプロセスサ部と論理回路部とから成る検証対象回路の例を示すブロック図である。

【図18】本発明の第4の実施の形態におけるコンパイル対象となる高級言語で記述されたプログラムの一例を示す図である。

【図19】本発明の第4の実施の形態における論理回路の可制限情報を示す図である。

【図20】本発明の第4の実施の形態におけるソフトウェアの検証画面を指定するソフトウェア指定情報を示す図である。

図である。

【図21】本発明の第4の実施の形態においてコンパイルが生成したメンバを示す図である。

【図22】本発明の第4の実施の形態においてコンパイルが生成した論理回路の制限情報を示す図である。

【図23】本発明の第1、第2、第3および第4の実施の形態におけるハードウェアの構成図の一例を示したブロック図である。

【図24】本発明の第1の実施の形態において論理回路の削減方法を示すフロー図である。

【図25】本発明の第2の実施の形態においてシミュレーション対象を交換する方法を示すフロー図である。

【図26】本発明の第3の実施の形態において制限情報抽出手段1002の概念フロー図である。

【符号の説明】

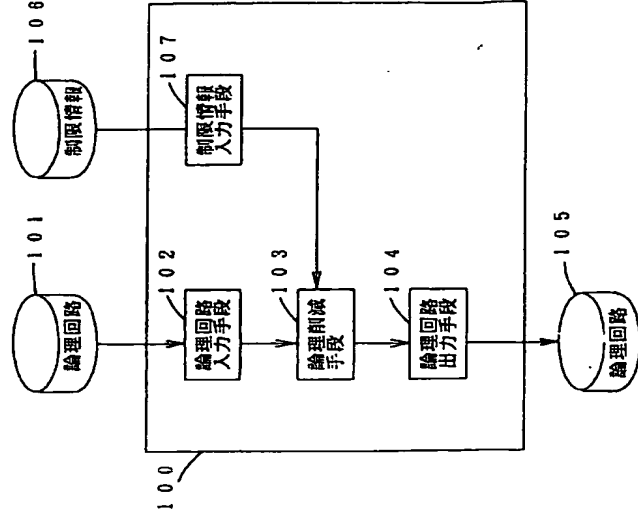
100 論理回路の変換装置

102 論理回路入力手段

103 論理削減手段

*

【図1】



【図2】

```

1 module filter(clk,reset,rm,we,rdata,wdata,sign,sgout,sgovf);
2   input clk;
3   input reset;
4   input re;
5   input we;
6   output [7:0] rdata;
7   input [7:0] wdata;
8   input [7:0] sign;
9   output [7:0] sgout;
10  output sgovf;
11
12  reg [7:0] mode;
13  reg [7:0] sghid1;
14  reg [8:0] sghid2;
15  reg [8:0] sghid3;
16  reg [8:0] sghid4;
17
18  always @(posedge clk) begin
19    if ( reset ) begin
20      mode = 7'b0;
21    end
22    else begin
23      mode <- we ? wdata : mode;
24    end
25  end
26
27  assign rdata = re ? mode : 8'haz;
28
29  always @(posedge clk) begin
30    sghid1 <- mode[0] ? 8'b0 : mode[4] ? sgin(7:0)
31      : ( 1'b0,sgin(7:1));
32    sghid2 <- sghid1 + mode[1] ? 8'h0 : mode[5] ?
33      sgin(7:0) : ( 1'b0,sgin(7:1));
34    sghid3 <- sghid2[7:0] + mode[2] ? 8'h0 : mode[6] ?
35      sgin(7:0) : ( 1'b0,sgin(7:1));
36    sghid4 <- sghid3[7:0] + mode[3] ? 8'h0 : mode[7] ?
37      sgin(7:0) : ( 1'b0,sgin(7:1));
38  end
39
40  assign sgout = sghid4[7:0];
41  assign sgovf = sghid2[8] | sghid3[8] | sghid4[8];
42 endmodule

```

【図3】

```

300 301 302 Flset ( ) ignore
303
304 filter.mode fix 8'hfa...303
305 filter.re fix 1'b0 -- 304
306 filter.sgovf ignore --305

```

【図20】

【図4】

【図7】

```

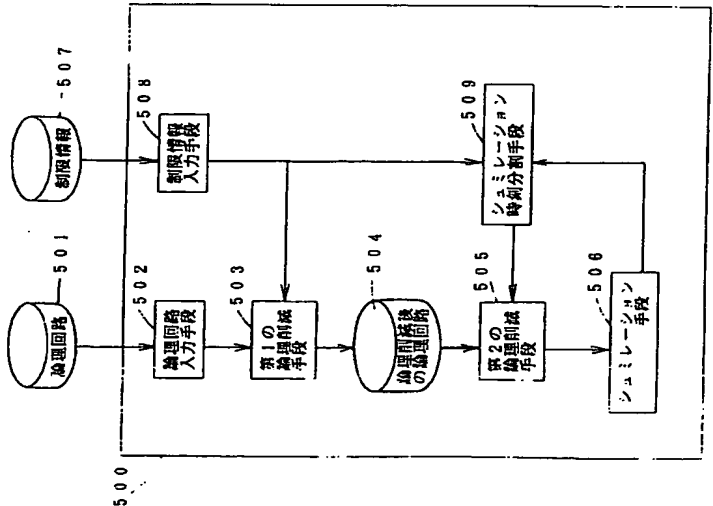
1 module filter(clk,reset,rm,we,rdata,wdata,sign,sgout,sgovf);
2   input clk;
3   input reset;
4   input re;
5   input we;
6   output [7:0] rdata;
7   input [7:0] wdata;
8   input [7:0] sign;
9   output [7:0] sgout;
10  output sgovf;
11
12  reg [7:0] mode;
13  reg [7:0] sghid1;
14  reg [8:0] sghid2;
15  reg [8:0] sghid3;
16  reg [8:0] sghid4;
17
18  always @(posedge clk) begin
19    if ( reset ) begin
20      mode = 7'b0;
21    end
22    else begin
23      mode <- we ? wdata : mode;
24    end
25  end
26
27  assign rdata = re ? mode : 8'haz;
28
29  always @(posedge clk) begin
30    sghid1 <- mode[0] ? 8'b0 : mode[4] ? sgin(7:0)
31      : ( 1'b0,sgin(7:1));
32    sghid2 <- sghid1 + mode[1] ? 8'h0 : mode[5] ?
33      sgin(7:0) : ( 1'b0,sgin(7:1));
34    sghid3 <- sghid2[7:0] + mode[2] ? 8'h0 : mode[6] ?
35      sgin(7:0) : ( 1'b0,sgin(7:1));
36    sghid4 <- sghid3[7:0] + mode[3] ? 8'h0 : mode[7] ?
37      sgin(7:0) : ( 1'b0,sgin(7:1));
38  end
39
40  assign sgout = sghid4[7:0];
41  assign sgovf = sghid2[8] | sghid3[8] | sghid4[8];
42 endmodule

```

【図7】

【図7】

【図5】



【図6】

```
module system;
.
.
.
filter filter1( .clk(clk), .reset(reset), .re(re1),
.we(we1), .rdata(rdata), .wdata(wdata),
.sigin(sigin1), .sigout(sigout1), .sigovf());
filter filter2( .clk(clk), .reset(reset), .re(re2),
.we(we2), .rdata(rdata), .wdata(wdata),
.sigin(sigin2), .sigout(sigout2), .sigovf());
assign sigout = mode ? sigout1 : sigout2;
.
.
endmodule;
```

【図9】

```
module system;
.
.
.
filter filter1( .clk(clk), .reset(reset), .re(re1),
.we(we1), .rdata(rdata), .wdata(wdata),
.sigin(sigin1), .sigout(sigout1), .sigovf());
assign rdata = 8'hzz;
assign sigout2 = 8'h28;
assign sigout = mode ? sigout1 : sigout2;
.
.
.
endmodule;
```

【図14】

```
1. 0x80 remove
2. {0x80} filter.mode
3. {0x80}.read filter.re
4. {0x80}.write filter.we
5. PSR & 0x00 filter.sigovf
{
1401
1402
1403
}
```

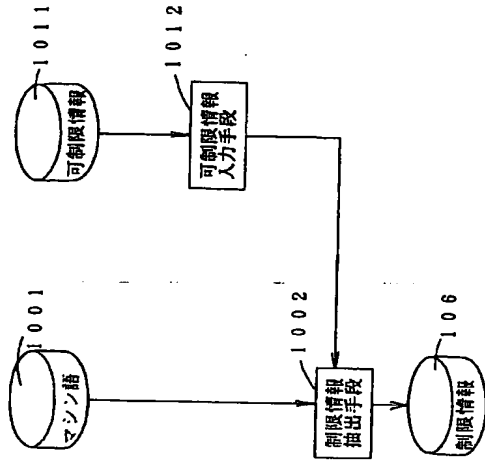
【図8】

```
1 module filter(clk, reset, re, ve, wdata, wdata, sdata, sdata, sigout, sigout);
2 input clk;
3 input reset;
4 input re;
5 input ve;
6 output (7:0) rdata;
7 input (7:0) wdata;
8 output (7:0) sigin;
9 output (7:0) sigout;
10
11 reg (7:0) rdata;
12 reg (7:0) sdata;
13 reg (8:0) sdata;
14 reg (8:0) sdata;
15 reg (8:0) sdata;
16 reg (8:0) sdata;
17
18 always @(posedge clk) begin
19   if (reset) begin
20     mode <- 7'h0;
21   end
22   else begin
23     mode <- ve ? wdata : sdata;
24   end
25 end
26
27 always @(posedge clk) begin
28   sdata1 <- mode[0] ? 8'h0 : mode[1] ? sdata[7:0]
29   : { 1'b0, sdata[7:1] };
30   sdata2 <- sdata1 + mode[1] ? 8'h0 : mode[5] ?
31   sdata[7:0] : { 1'b0, sdata[7:1] };
32   sdata3 <- sdata2[7:0] + mode[2] ? 8'h0 : mode[6] ?
33   sdata[7:0] : { 1'b0, sdata[7:1] };
34   sdata4 <- sdata3[7:0] + mode[3] ? 8'h0 : mode[7] ?
35   sdata[7:0] : { 1'b0, sdata[7:1] };
36 end
37
38 assign sigout = sdata4[7:0];
39 endmodule
```

【図15】

| | | | | |
|------------------|--------|------|---------|-------|
| 1. filter.mode | fix | hold | PC=L1 | PC=L8 |
| 2. filter.sigout | ignore | | initial | PC=L8 |
| 3. filter.re | fix | hold | initial | PC=L8 |
| | { | { | { | { |
| 1500 | 1501 | 1502 | 1503 | 1504 |

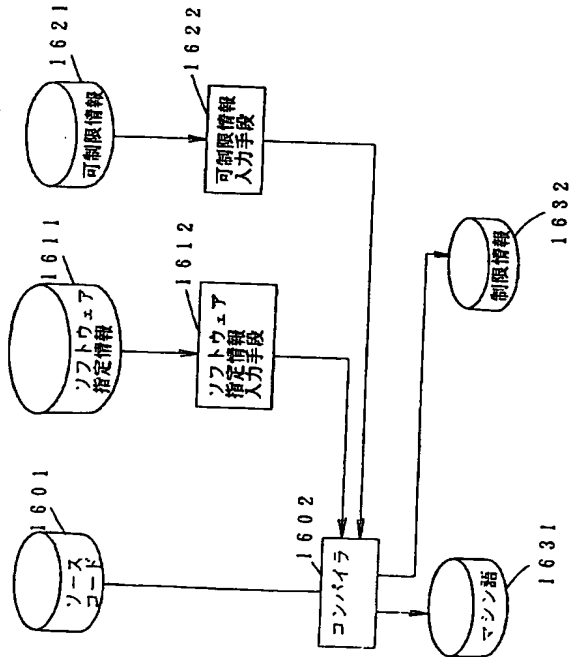
【図10】



【図11】

| 命令 | デスティネーション | ソース |
|------|---------------------|---------------------|
| LD | regA or regB | アドレス |
| ST | regA or regB | アドレス |
| MOV | regA or regB or PSR | regA or regB or 定数値 |
| ADD | regA or regB | regA or regB or 定数値 |
| SUB | regA or regB | regA or regB or 定数値 |
| MUL | regA or regB | regA or regB or 定数値 |
| CMP | regA or regB | regA or regB or 定数値 |
| BEQ | ラベル | |
| JMP | ラベル | |
| RETI | | |
| NOP | | |
| HALT | | |

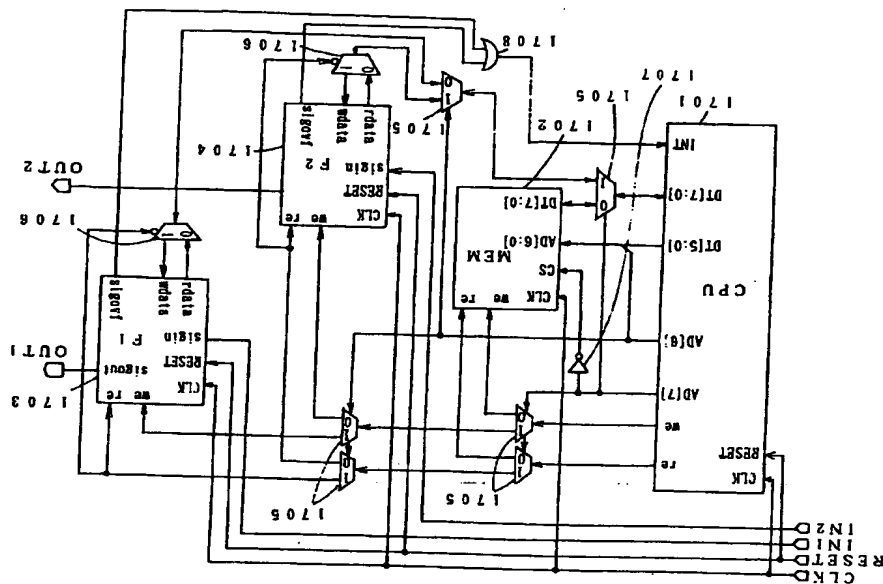
【図16】



【図19】

| | | |
|------------------|-----------------------|--------|
| 1. 0xC0 | F1 | remove |
| 2. 0x60 | F2 | remove |
| 3. (0xC0) | F1. mode | |
| 4. (0x80) | F2. mode | |
| 5. (0xC0). read | F1. re | ignore |
| 6. (0xC0). write | F1. we | ignore |
| 7. (0x80). read | F2. re | ignore |
| 8. (0x80). write | F2. we | ignore |
| 9. PSR & 0x00 | F1.sigovf, F2.sigovf. | ignore |
| } | | 1901 |
| } | | 1902 |
| } | | 1903 |

【図17】



(図18)

```
Flset(void)
{
    char a=0xfa;
    memset(0xc0, a, 1);
}

F2set(void)
{
    char a=0xbc;
    memset(0x80, a, 1);
}

main( )
{
    Flset();
    F2set();
    for(a=0x100; a!=0 : a--
    ;
```

(図22)

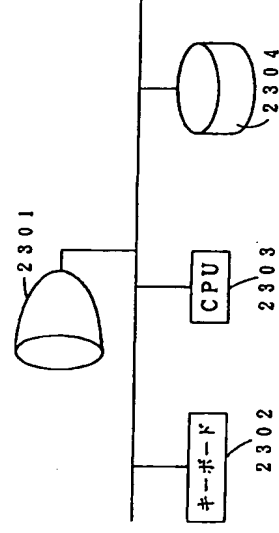
```

F1      remove      hold      PC=L1
F2.mode fix         ignore    initial
F2.sigovf ignore    hold      initial
F2.re   fix         hold      initial
        }           }
        2200        2201      2202      2203
```

(図21)

| | | | |
|----|------|-------|--------|
| L0 | MOV | 0xbc | regA |
| L1 | ST | regA | (0x80) |
| L2 | MOV | 0x100 | regA |
| L3 | MOV | 0x00 | regB |
| L4 | SUB | regA | -1 |
| L5 | CMP | regA | regB |
| L6 | BEQ | L8 | |
| L7 | JMP | L4 | |
| L8 | HALT | | |

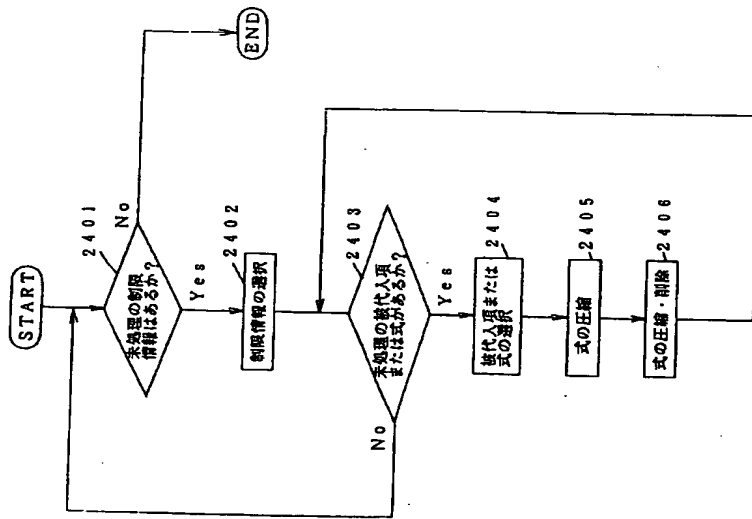
(図23)



(23)

特開2001-22808

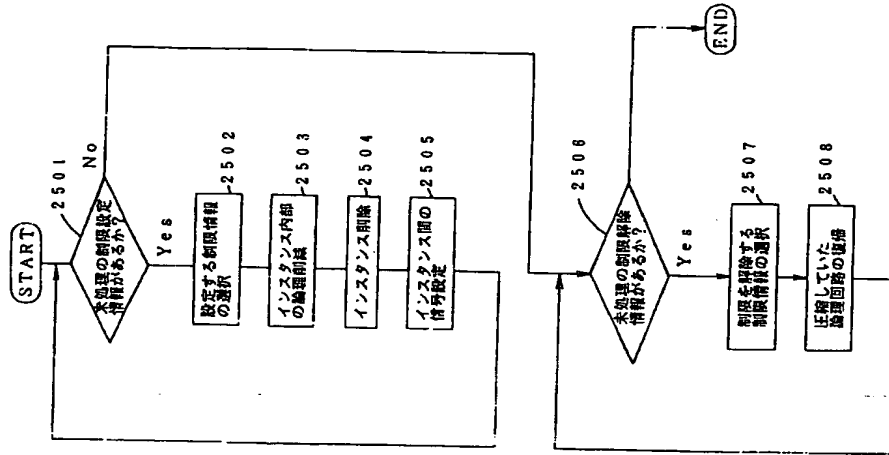
【図24】



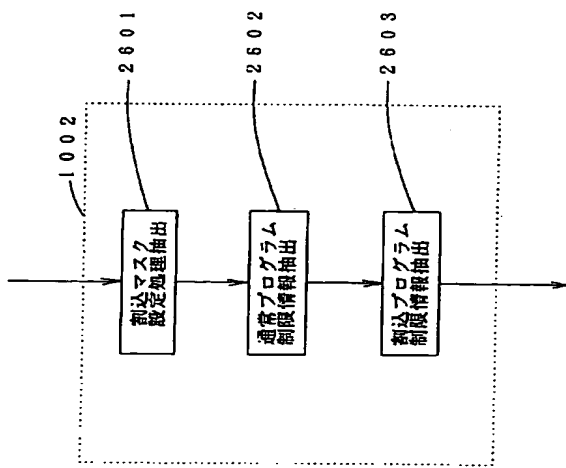
(24)

特開2001-22808

【図25】



〔図26〕



THIS PAGE BLANK (USPTO)